A Case For Lightweight Dynamic Event Based Monitoring and Management Support For Large Scale DataCenters

Mahendra Kutare
Center for Experimental Research in Computer System
College of Computing
Georgia Institute of Technology, Atlanta, GA 30318, USA
imax@cc.gatech.edu

ABSTRACT

Future large scale systems, such as cloud datacenters, with increased core counts will soon result in infrastructures with millions of cores. This poses challenges for monitoring and management not yet met by existing experimental or commercial software systems. At the core of this challenge is to perform continuous and on-demand monitoring queries over distributed aggregated data resulting from distributed monitoring data streams. Of particular importance is the ability to quickly detect, correlate and react to system issues.

Our research is developing monitoring and management methods and infrastructure that can scale and also exhibit small lag. Our approach is to use event based system design and distribute monitoring, select data aggregation and analysis and actuation across datacenter subsystems and machines. Finally, by embedding management into the underlying system and management infrastructure, based on modern support for virtualization, management is separated from applications, enabling its usage and change without affecting user codes.

1. PROBLEM DESCRIPTION

Existing monitoring and management systems with single or multiple hierarchy for monitoring data aggregation and centralized analysis and coordination, are not designed to scale for future large scale systems. Some of the key challenges for monitoring and management systems are -

1. Ability to proactively detect, correlate and react to system issues.
2. Ability to change part of monitoring data path due to changes in monitors, systems, subsystems or platforms.
3. Exhibit small lag in reacting to system issues caused due to external or internal factors affecting the systems.
4. Global and hierarchical correlated view of the information at an acceptable overhead and within an allowable precision.
5. Distributed coordination among various hierarchies of aggregated data, monitors and actuators across the environment.

In particular, for large scale systems concerning our research such as future datacenters, key trends are shaping future technologies. First, there is an inexorable move toward many-core chips, which are increasingly composed of both general purpose cores and those specialized to certain tasks. Coupled with increased blade server densities and hardware disaggregation, these result in numbers of end systems and a degree of heterogeneity that makes it imperative to intimately integrate facilities for online and automated management into such systems. Second, increased demand for ever larger and more reactive datacenters, in part driven by cloud computing, will lead to scales of millions of cores, making it critically important for automated management methods to scale. It also implies the need for them to exhibit basic properties that include extensibility, the ability to interact with diverse management subsystems, and robustness. Third, virtualization is a becoming a necessary element of any study in systems management, in part because of its known benefits like server consolidation in datacenter environments.

Thus two most important categories of system issues for monitoring and management systems focused in our research concerning future datacenter environment are -

1. *Dynamism*
   With the growing complexity of various groups, applications and services in particular importance to the datacenter management is the ability to quickly detect and react to system issues, in order to mitigate and contain effects deleterious to application performance or datacenter health. The monitoring and management infrastructure should exhibit small lag under changes occuring due to dynamic and distributed virtualized datacenter environment.
2. *Scalability*
   The traditional approaches using centralized and reactive techniques for monitoring, data aggregation and analysis and actuation across datacenter subsystems and machines will not be able to scale to millions of core. Each of these components will have to be distributed and thus sheer scale introduces a problem of coordinated distributed decision across the datacenter environment. The monitoring and management mechanisms needs to scale in and scale out efficiently.

2. IMPORTANCE

The importance of the lightweight, distributed and proactive monitoring and management system can be understood from the application scenarios in particular for future datacenter environment.

In current datacenter environment, monitoring and management systems provides hardware and software based approach to the monitoring. In both approaches, monitoring data is collected about systems and subsystems of interest through monitoring agents, then passed to centralized management servers for data collection, filtering, correlation and root-cause analysis. The actuation mechanisms provided with these infrastructure are primarily user defined scripts. The central point of analysis, correlation, actuation in current infrastructure will not be able to respond to system issues quickly enough.

For example, lets say we want to monitor amount of disk space used on specified directory or logical driver. We would then select a logical drive or filesystem size for monitoring. Say we select /usr/local filesystem which can hold 150 MB of data. We can then set up a monitor to watch for this size and run a user-created script that deletes log or backup files created by web application that are more than one day old whenever the filesystem contains 135 MB of data.

For future datacenters as mentioned above, to manage scale, we will require distributed hierarchical aggregation of monitoring data which brings the problems of distributed hierarchical analysis, correlation, actuation and lag between these components.

With our example, we will require correlation among disk space usage monitoring data occuring at potentially thousands of cores at various levels of monitoring hierarchy and perform analysis and actuations potentially at those hierarchy levels. We can see how traditional central point of analysis, correlation and actuation not designed for large scale will exhibit lag in analysis, decision making and actuation induced due to sheer scale.

Since the current systems do not provide distributed real-time monitoring data correlation from a variety of different systems, subsystem and platforms with in-built coordination and actuation mechanisms they do not proactively respond to critical issues resulting in downtime which can save millions of dollars in potential lost revenue. For example, the Data Center Journal reports the cost of downtime continues to climb, with some industries approaching **$3 million per hour in lost revenue due to issues related to downtime**. We contend that the lag exhibited between monitoring, detecting issues to mitigating them, will become a key issue causing catrastrophic failures for future datacenter infrastructure.

3. EXISTING SOLUTIONS

Most common commercial monitoring and management systems are HP iLO [1] and IBM Tivoli [2]. These systems as mentioned before perform centralized analysis and actuation with scripts based triggering mechanism and are primarily request/response based systems. They provide system defined and user specified metrics monitoring but lack support for run time changes to monitoring, monitoring and mangement hierarchy for analysis and correlation and further support for dynamic changes to monitoring and management.

Several academic efforts in areas of distributed systems meet the above mentioned challenges in system design partially. These efforts ranges from distributed monitoring systems, data aggregation systems and on-demand monitoring operations in large scale systems. All these efforts though applicable in large scale datacenter environment provide partial solutions to key challenges detailed above. Astrolabe [10] provides a generic aggregation abstraction using single static tree, SDIMS [12] deploys multiples trees for number of metrics using single group for the entire system while Moara [11] optimizes multiple group based aggregation trees.

Distributed monitoring systems such as Ganglia [8] uses single hierarchy and collects all the data centrally. It lack support for in-network aggregation or run time transformation of monitoring events and data path. MON [6] provides support for one-shot queries and constructs queries on-demand for large scale infrastructures but also lack support for run time transformation of monitoring events and data paths, distributed coordination and actuation across monitoring and data aggregation hierarchy.

None of the above systems meet the key challenges described for future large scale system and in particular for future datacenter evironment concerning our research. They do not provide a holistic, dynamic view of the entire datacenter, i.e., distributed correlation and actuation which limits the effectiveness of these solutions. In particular, the ability to quickly detect and react to system issues at multiple levels and cross cutting responsibilities with multiple hierarchies in a datacenter. Almost all of the above systems, are based upon request/response paradigm which we contend will not scale for the datacenter sizes we described above.

On the other hand, there is large body of research using event based systems in internet scale systems. Stream processing systems such as Aurora [4] and Borealis [3] combine event-based, push based dissemination with transformation capabilities. Hermes [9] provides event based middleware with advanced features of composite event detection, type and attribute based event routing with programming language integration. Our own research efforts Echo [5] and iFlow [7] on event-action and stream based systems are used successfully in high performance and enteprise domains.

These event based systems though not build to handle the problems concerning our research but provide solid design principles and plaform to build large scale systems in particular datacenter environment concerning our research.

4. POTENTIAL APPROACHES

Our approach rejects the request/response model of monitoring used in some commercial systems [2], [1] in favor of an event-action based model that triggers events of interest based on low level 'active' sensors, the latter able to recognize simple trigger conditions using small-scale local states. At the next higher level, monitoring actions are structured

as event-action overlays distributed across logical and physical subsystems.

Our work builds further on our past research efforts Echo [5] and iFLOW [7] in context of event-action and stream based systems.

For distributed monitoring, our approach is to build event based systems along with dynamic code generation providing a way to inject various types of filters and transformers at run time on monitoring data delivery path. It thus provides a flexible and scalable way to monitor and analyse as close as possible to the source for obvious benefits of reduced network overheads.

Specifically, by distributing monitoring, select data aggregation and analysis, and actuation across datacenter subsystems and machines we localize problem detection and mitigation and reduce data volumes via local data analysis; by making such distribution dynamic, i.e., deploying management code as and when needed, management actions can focus on subsystems or applications that currently require them, triggered by continuous, lightweight monitoring concerned primarily with basic system or application health. Finally, by embedding management into the underlying system and management infrastructure, based on modern support for virtualization, management is separated from applications, enabling its usage and change without affecting user codes.

To handle coordination across various data aggregation hierarchies, we intend to investigate composite event detection approach of event based systems to provide generic support for online detection and correlation of events at various levels of single hierarchy and across multiple hierarchies thus providing holistic and unified perspective.

5. ONGOING WORK AND CONCLUSION

Our research is building the Cyton system, using which the capture of monitoring data, its aggregation, and analysis can be performed with the low overheads required to enable continuous, lightweight, software-based monitoring of virtualized systems and the applications using them. Specifically, using the Xen hypervisor, Cyton's dynamic monitoring functionality is embedded into control domains that are separated from those running application codes. Continuous monitoring is used to detect abnormal system or application behaviors, whereupon additional, more substantive monitoring functions are triggered. This enables the runtime problem diagnosis and mitigation that is the goal of our research. The next step in this work is to demonstrate that the lightweight nature of this system coupled with its dynamic capabilities enables it to scale to thousands of nodes with acceptable overhead.

6. ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Karsten Schwan, research scientists Dr. Greg Eisenhauer and Dr. Matthew Wolf for guiding and HP researchers – Dr. Vanish Talwar, Dr. Parthasarathy Ranganathan and Dr. Niraj Tolia for providing insightful comments and inputs to this work.

7. REFERENCES

[1] Hp iLO - http://h18000.www1.hp.com/products/servers/management/ilo/.

[2] IBM Tivoli - http://www-01.ibm.com/software/tivoli/.

[3] M. Balazinska, H. Balakrishnan, S. Madden, and M. Stonebraker. The Design of the Borealis Stream Processing Engine. *In Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 13–24, January 2005.

[4] D.Abadi, D.Carney, U.Cetintemel, et al. Aurora: A New Model and Architecture for Data Stream Management. *VLDB*, 12(2):120–139, August 2003.

[5] G. Eisenhauer, F. Bustamente, and K. Schwan. Event Services for High Performance Computing. *In Proceedings of High Performance Distributed Computing (HPDC)*, August 2000.

[6] J.Liang, S.Y.Ko, I.Gupta, and K.Nahrstedt. MON: On-demand Overlays for Distributed Systems Management. *In Proceedings of the 2nd USENIX Workshop on Real, Large Distributed Sytems (WORLDS)*, 2005.

[7] V. Kumar, Z. Cai, B. F. Cooper, G. Eisenhauer, K. Schwan, et al. iFLOW: Resource-aware Overlays for Composing and Managing Distributed Information Flows. *Eurosys*, 2006.

[8] M.L.Massie, B.N.Chun, and D.E.Culler. The Ganglia Distributed Monitoring System: Design, Implementation and Experience. *Parallel Computing*, 30(7), July 2004.

[9] P. R. Pietzuch. *Hermes: A Scalable Event-Based Middleware*. PhD thesis, Computer Laboratory, Queens' College, University of Cambridge, February 2004.

[10] R.V.Renesse, K.P.Birman, and W.Vogels. Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management and Data Mining. *ACM Transactions on Computer Systems*, 21(2):164–206, May 2003.

[11] S.Y.Ko, P. Yalagandula, I.Gupta, V.Talwar, D.Milojicic, and S.Iyer. Moara: Flexible and Scalable Group Based Querying Systems. *In Proceedings of the 9th ACM/IFIP/USENIX Middleware*, 2008.

[12] P. Yalagandula and M. Dahlin. SDIMS: A Scalable Distributed Information Management System. *In Proceedings of ACM SIGCOMM*, 2004.